ICOOL: A TOOL FOR MUON COLLIDER SIMULATIONS

R.C. Fernow[*]
Bldg. 901A, Brookhaven National Laboratory, Upton, NY 11973

Abstract

Current ideas for designing neutrino factories [1,2] and muon colliders [3] require unique configurations of fields and materials to prepare the muon beam for acceleration. This so-called front end system must accomplish the goals of phase rotation, bunching and cooling. We have continued the development of a 3-D tracking code, ICOOL [4], for examining possible muon collider front end configurations. A system is described in terms of a series of longitudinal regions with associated material and field properties. The tracking takes place in a coordinate system that follows a reference orbit through the system. The code takes into account decays and interactions of ~50-500 MeV/c muons in matter. Material geometry regions include cylinders and wedges. A number of analytic models are provided for describing the field configurations. Simple diagnostics are built into the code, including calculation of emittances and correlations, longitudinal traces, histograms and scatter plots. A number of auxiliary codes can be used for pre-processing, post-processing and optimization.

1. Program overview

The design of the front end of a muon collider faces a number of difficulties not ordinarily encountered in accelerator design. By “front end” we refer specifically to the region between the pion collection system and the start of the accelerator chain. This region typically contains the following components

- phase rotation to reduce the energy spread of the beam
- pion to muon decay drift
- bunching
- muon cooling
- emittance exchange

Phase rotation makes use of induction linacs or low frequency *rf* cavities. Bunching is done with a series of *rf* cavities and drifts. The only known method that can cool the beams in a time comparable to the muon lifetime is ionization cooling [5,6]. This method requires directing the particles in the beam at a large angle through a low Z absorber material in a strong focusing magnetic channel and then restoring the longitudinal momentum with an *rf* cavity. Emittance exchange is the only practical method for reducing the longitudinal emittance of the muon beam and involves passing the beam through wedge-shaped absorbers in the presence of dispersion. Thus beam interactions in matter, solenoidal focusing channels, and dispersive lattice elements all need to be included in the simulations.

A major part of the simulation effort for the Muon Collaboration has been directed toward the development of two programs to accomplish these goals. The first, ICOOL, provides the flexibility

[*] Email: fernow@bnl.gov

to quickly examine widely different ideas for configuring muon collider front ends. For example, setting up desired field configurations is accomplished in ICOOL using predefined, analytic field models. This simplifies the adjustment of parameters for the field to obtain some desired result. The second set of programs, DPGeant and Geant4 [7], are based on the GEANT code system [8]. It is possible to describe very complicated 3-D problem geometries using these codes and to calculate quantities to greater accuracy than with ICOOL. Geant4 typically gets its field distributions from maps generated by other, more accurate field computation codes. In our experience the two codes have been quite complementary. One program is frequently used to check results from the other.

ICOOL uses a command input file consisting of five parts:

- simulation control variables
- beam definition
- control of physics interactions
- diagnostics
- region definition

Current ideas for muon collider front ends make extensive use of solenoidal channels. For this reason we define a region to encompass a cylindrical volume which has a fixed length along the reference orbit. A region can be subdivided radially in up to 5 subregions. Each subregion has a field type, material type, and material geometry associated with it. Particles are allowed to pass back and forth between radial subregions. Wedge shaped material geometries are provided for reducing the momentum spread in dispersive regions. There is no practical limit on the number of regions in a problem. After validity checking, the region data are stored in a direct access disk file.

The program tracks the particles to the end of a given region and generates any desired diagnostics. It then continues to track the surviving particles to the end of the next region. This program structure was adopted to make it possible to eventually add space charge interactions. At present the code only has a very crude space charge model.

The region description language has two looping structures to aid in describing complicated, repetitive systems. A group of regions, such as *rf* cavity cells, may be repeated as often as desired using a REPEAT structure. Groups of REPEAT structures and isolated regions may be combined into a CELL structure, which may also be repeated as often as desired. In addition a CELL has its own field type associated with it. This allows applying a background solenoid field, for example, over a sequence of regions, each of which has its own local field.

In addition to the physical regions described above, the user can insert “pseudoregions” into the command file at various locations to accomplish tasks, such as forcing diagnostic output, collimating the beam, transforming the beam with a TRANSPORT element, redefining the reference particle, etc.

The program can initialize the phases of long strings of *rf* cavities by using an on-axis reference particle. The most commonly used algorithm tracks the reference particle through absorbers and other non-cavity regions, taking into account the mean energy lost there. The energy of the reference particle is increased in *rf* cavity regions by assuming the particle gains a constant energy per unit length. It is then possible to calculate the time the reference particle passes the center of each cavity and to adjust the cavity electric fields to be at zero crossing at these times. After the relative cavity phases have been determined, the user can control the beam's interaction with the *rf* fields by adjusting the mean launch time of the particles in the beam or by applying additional phase shifts to individual cavities.

For maximum compatibility across the Muon Collider collaboration it was decided to write the code in Fortran 77 and to restrict graphics inside the program to simple character based "printer plots". The program has been run successfully on UNIX, PC, and Macintosh platforms.

A simulation of transverse cooling for the Feasibility Study 2 neutrino factory [2] using the sFOFO alternating solenoid configuration takes 122 min on a 500 MHz Pentium PC. This simulation involved tracking 2400 muons through 108 m. There were 1169 regions consisting of liquid hydrogen absorbers with aluminum windows and pillbox *rf* cavities with beryllium windows, all immersed in a tapered, alternating direction, periodic solenoidal field lattice.

2. Program components

2.1 Particle generation

The user can choose to generate gaussian or uniform initial particle distributions inside the program. Available particles are electrons, muons, pions, kaons and protons of either sign.

It is possible to generate a number of important correlations in the initial beam distribution. (1) The beam can be given an initial angular momentum that is appropriate for starting the simulation inside a solenoidal field. (2) The beam can have a correlation between its forward velocity and its transverse position and/or divergence. This is necessary to counteract the bunch lengthening from path length differences encountered in a long solenoid. (3) The longitudinal phase space can be prepared in an *rf* bucket that matches the requirements of a subsequent *rf* cavity. (4) The transverse phase space can be initialized with an arbitrary set of Twiss parameters $\{\alpha, \beta, \varepsilon\}$ for matching into the subsequent lattice.

The initial beam specification is quite flexible, allowing a mixture of particle types, each with its own phase space distributions and correlations. The input beam information may also be read in from an existing set of particle data on an external file. There is no practical limit on the number of particles. Currently the first 50,000 particles are stored in memory and particles in excess of this are stored on a direct access disk file. The program can save the particle state after any region in the same format required for input, so problems can be run in stages

2.2 Electromagnetic fields

Typically the user chooses one of 20 predefined field types as the region field. There are usually several models for each type of field configuration representing different approximations or symmetry assumptions. Usually one of these models, e.g. a constant B_z solenoid region, is simple enough so that the particle's interaction with the field may be readily understood. Other more complicated models that satisfy Maxwell's equations can be used for more accurate modeling, e.g. the solenoidal field from a cylindrical current sheet. If the superposition of a region and a cell field is not sufficient, a background field can be defined on a 3-D grid. This field can be built up from any combination of the predefined fields.

Sometimes the fringe field of a region has not fallen to 0 at the defined end of the region. For this reason, an option exists to include in the field of a given region contributions from the fields in neighboring regions.

Simulations of emittance exchange have used dipoles, bent solenoids, or helices to generate the required dispersion, together with wedge shaped absorbers for removing the momentum spread. ICOOL uses analytic models for these fields. Linear and polynomial shaped wedge regions are provided in the program.

The following function is widely used in the code to give a continuous expression for a magnetic field or vector potential component on the axis.

$$f(s) = \frac{1}{2}\left[\tanh\left(\frac{s-E}{\lambda}\right) - \tanh\left(\frac{s-C-E}{\lambda}\right)\right]$$

where E is the distance of the leading edge of the field pattern from the start of the region, C is the width of the central “high field” part of the distribution, and λ gives the rate of fall off at the ends. Other field components on and off the axis are calculated from Maxwell’s equations. One nice feature of this function is that the derivatives can also be calculated analytically to any order. Derivatives up to seventh order are used in the code.

2.3 Interactions and decays

Muon and pion decays can be simulated. An option exists to continue tracking the charged daughter product from a decay. Energy loss, straggling and scattering in matter are simulated as continuous processes. The code currently uses a set of 12 internally defined elements and compounds that are of greatest interest for ionization cooling. The mean value of the energy loss is computed using the Bethe-Bloch formalism including the density effect. Fluctuations in the energy loss may be sampled from Gaussian, Landau or Vavilov distributions [8]. Sampling from the Vavilov distribution is switched to a Gaussian or Landau distribution if the local values of the stepsize and velocity require it.

Multiple Coulomb scattering may be sampled from Gaussian distributions with widths determined by the Rossi-Greisen, Highland or Lynch-Dahl models, from Rutherford single scattering or from the Moliere distribution [8,9]. Some additional user control is provided for the parameters used in the Moliere theory, since the amount of scattering is very important in ionization cooling and there is some uncertainty in the application of Moliere theory to low Z materials. Sampling from the Moliere distribution is switched to plural Rutherford scattering if the stepsize is so small that the effective number of scatters is less than 20. Delta ray events with correlated angle-energy loss variables may also be simulated. Nuclear interactions, pair production and bremsstrahlung are not simulated.

2.4 Tracking

Since the system is defined in terms of regions along the reference trajectory (s), the independent variable in the equations of motion is *ds*. The dependent variables are the transverse positions, momentum, and the polarization. The position and momentum equations are defined up to third order in the deviations from the reference orbit [10]. The local value of the curvature is computed for both transverse planes. Particle stepping can be done using fixed steps or by using an adaptive stepsize algorithm. The adaptive algorithm takes into account constraints due to the validity of the multiple scattering and straggling models and due to the local gradient of the field. Particle stepping is done using fourth-order Runge-Kutta integration [11].

2.5 Diagnostics

ICOOL has a number of internal diagnostics, which can be used to check if the simulation is working properly. Kinematic variables and field values can be plotted for individual particles as a function of distance (Z-history). The mean, standard deviation, minimum and maximum values for the ensemble of particles can be plotted for any variable at the end of each region (R-history). In addition the program has a number of statistical analyses. Any defined variable can be histogramed at the end of any region. Scatter plots may be made of any variable at one region with any variable at the same or any other region. Moments are computed for each of the histogram and scatter plot variables. Emittances, polarization values and/or covariance matrices may be computed after any region.

Besides the built-in diagnostics, the program can write out files of particle and field information. These files can be generated (1) after specified regions, (2) automatically at the end of every region, or (3) repeatedly after a specified number of steps inside a region. This information can be used for more sophisticated analysis or plotting in a separate post-processing program. Other diagnostic files can be written giving the magnetic field grids or the *rf* phases generated by the program.

3. Auxilliary codes

Two preprocessor codes have been written to simplify the process of setting up a complicated ICOOL problem. For example, the front end of the neutrino factory in Feasibility Study 2 [2] contained 1466 separate regions. In the process of specifying such a complicated system, it is very easy to make mistakes. NIME [12] is a Tcl/TK script that accepts an ASCII input file. The user can define macros with parameters corresponding to needed ICOOL commands along with copious explanatory comments. The program expands the macro commands in the file to produce an ICOOL command file input. XICOOL [13] also provides a convenient method for setting up a problem by enabling multiple levels of parameterized text macros and allowing for comments. In addition it provides a means of generating all associated files necessary to run a problem in a single configuration file.

Almost every ICOOL user has developed their own post-processing file for analyzing and plotting the results of interest from the simulations. This is probably inevitable since each user has a particular physics interest and a favorite set of graphics tools available in their computing environment. All these post-processing programs start by reading the ICOOL post processor output file[1]. One emittance computation program in wide use is ECALC9 [14]. This code computes emittances for symmetric solenoidal channels from the determinant of the covariance matrix. It also computes modified Courant-Snyder parameters that take angular momentum into account [15].

[1] This file is named FOR009.DAT in ICOOL.

A third type of auxiliary code involves using ICOOL inside an optimizer. Any such code must be capable of specifying the variables to be modified and of defining a merit function for the results of the optimization. OPTICOOL [16] is one such program that has the capability of distributing jobs across a UNIX farm. Amoeba [17] is a simpler program that has been used to design the coil configuration for lattice matching regions.

4. Example results

As an example of a cooling system that has been modeled using ICOOL, consider the alternating solenoid lattice [3]. One 2 m long cell in this lattice is shown in Fig. 1.

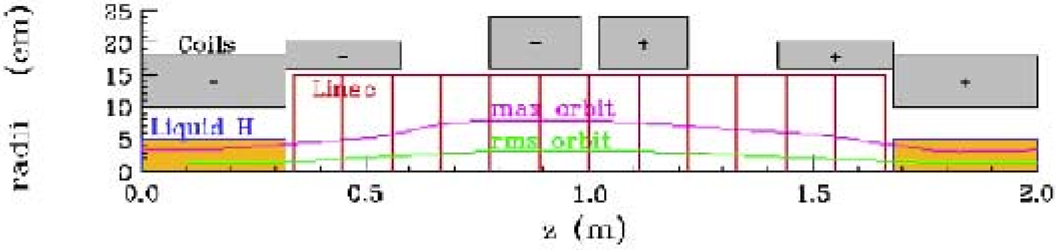

Figure 1. One cell of an alternating solenoid cooling lattice [3].

The alternating-direction solenoidal field is generated by the 6 coils. The coils surround liquid hydrogen absorbers at both ends and 12 pillbox *rf* cavity cells in the center. The direction of the field changes in the middle of the *rf* cavity. The change in transverse emittance passing down a 24 m long lattice of these cells is shown in Fig. 2.

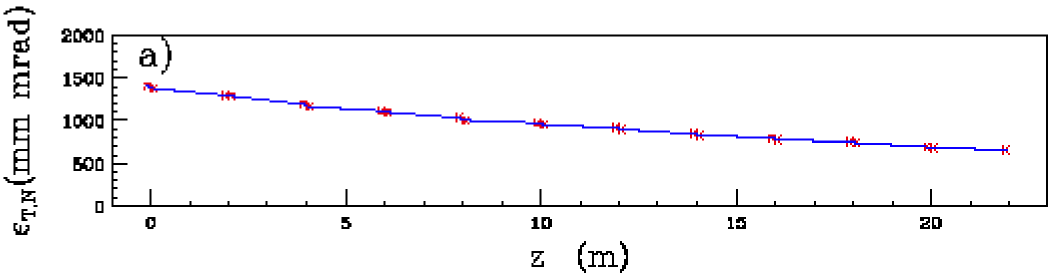

Figure 2. Transverse normalized emittance in the alternating solenoid lattice [3].

Acknowledgements

I would particularly like to thank R.B. Palmer, J.C. Gallardo, P. Lebrun, and D. Neuffer for their helpful suggestions. This research was supported by the U.S. Department of Energy under Contract No. DE-AC02-98CH10886.

References

[1] N. Holtkamp & D. Finley (eds), A feasibility study of a neutrino source based on a muon storage ring, Fermilab, 15 April 2000.

[2] Muon Collaboration, Feasibility Study 2 on a neutrino factory at BNL.

[3] C. Ankenbrandt et al, Status of muon collider research and development and future plans, Phys. Rev. Special Topics 2:081001-(1-73), 1999.

[4] R. Fernow, ICOOL: a simulation code for ionization cooling of muon beams, Proc. 1999 Particle Accelerator Conference, A. Lucio & W. MacKay (eds), p. 3020.

[5] A.N. Skrinsky & V.V. Parkhomchuk, Methods of cooling beams of charged particles, Sov. J. Part. Nucl. 12:223-247, 1981.

[6] D. Neuffer, Principles and applications of muon cooling, Part. Acc. 14:75-90, 1983.

[7] Most of the DPGeant development was done by P. Lebrun at FNAL.

[8] CERN Program Library, GEANT 3.2.1 Detector description and simulation tool, 1993.

[9] Particle Data Group, Review of particle properties, Eur. Phys. J. C 3:144-151, 1998.

[10] M.F. Tautz, Third order aberrations for ion orbits in static magnetic fields with mid-plane symmetry, Nuc. Instr. Meth. 84:29-36, 1970.

[11] W.H. Press et al, Numerical Recipes in Fortran, 2nd ed., Cambridge University Press, 1992, p. 704-716.

[12] W. Fawley, Quick user manual for the NIME pre-processor, http://www.cap.bnl.gov/mumu/software/nime.html

[13] S. Bracker, The XICOOL preprocessor for ICOOL input files, NuMu note 196, 2001.

[14] G. Penn, LBNL.

[15] G. Penn & j. Wurtele, Beam envelope equations for cooling of muons in solenoid fields, Phys.Rev.Lett. 85:764-7, 2000.

[16] S. Bracker, OPTICOOL – a multiprocessor optimizer for muon cooling simulation codes, NuMu note 197, 2001.

[17] R. Palmer, BNL.